

Marketplace Technical Foundation

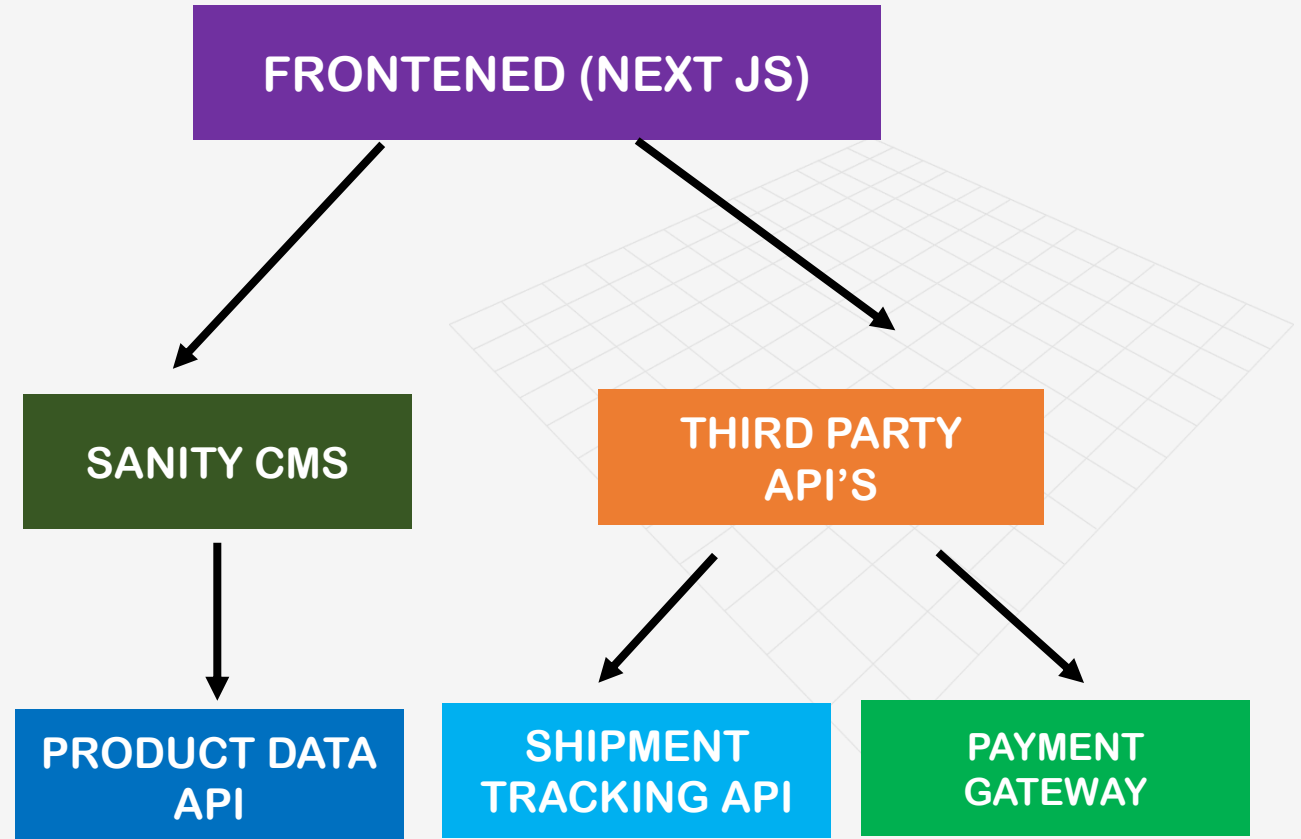
Rocket Seater Furniture

Launched By

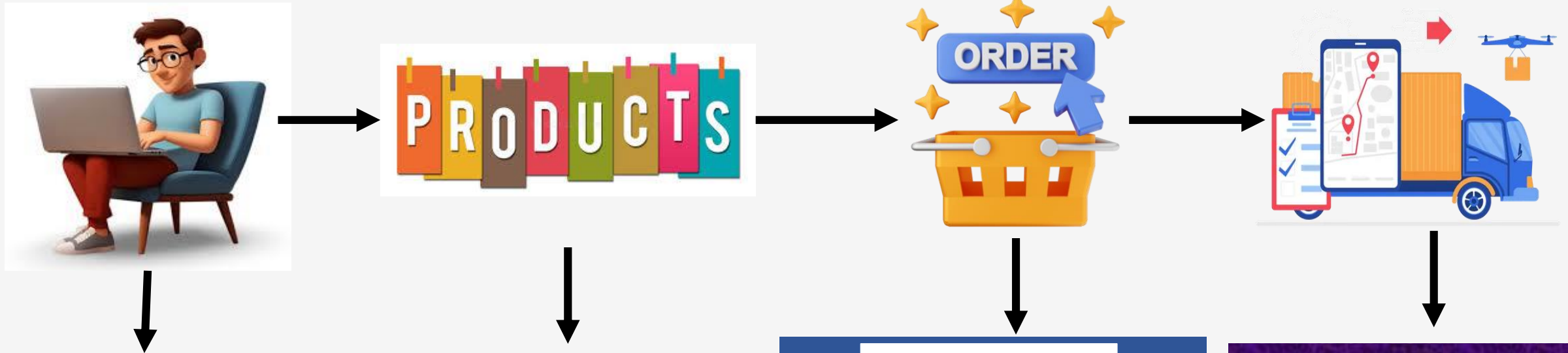
SABIHA SULTANA

Rocket Seater Furniture (System Architecture)

- **Frontend (Next.js):** Handles user interface and interactions
- **Sanity CMS:** Manages product data, customer details, and order records
- **Third-Party APIs:** Provide shipment tracking and payment processing
- **Product Data API:** Fetches product information from Sanity CMS
- **Shipment Tracking API:** Provides real-time shipment status updates
- **Payment Gateway:** Processes secure transactions



ROCKET SEATER FURNITURE (Key Workflows)



User Registration

- ❖ User enters registration details on frontend
- ❖ Data is sent to Sanity CMS for storage
- ❖ Confirmation is sent back to the user

Product Browsing

- ❖ User navigates to the shop page
- ❖ Frontend requests product data from Sanity CMS
- ❖ Products are displayed on the frontend

Order Placement

- ❖ User adds items to the cart
- ❖ User proceeds to checkout
- ❖ Order details are saved in Sanity CMS
- ❖ Payment is processed through the Payment Gateway
- ❖ Order confirmation is displayed to the user

Shipment Tracking

- ❖ User navigates to the shop page
- ❖ Frontend requests product data from Sanity CMS
- ❖ Products are displayed on the frontend

Rocket Seater Furniture (API REQUIREMENTS)

Product API



- Endpoint Name:/products
- Method: GET
- Description: Fetch all available products from Sanity
- Response Example:

```
{ "customerId": 123, "items": [{"productId": 1, "quantity": 2 }, { "productId": 5, "quantity": 1 } ], "totalAmount": 1497, "shippingAddress": "123 Main St, City, Country" }
```

ORDER API



- Endpoint Name: /orders
- Method: POST
- Description: Create a new furniture order in Sanity
- Response Example:

```
{ "orderId": "ORD12345", "status": "confirmed", "estimatedDelivery": "2023-05-25" }
```

SHIPMENT TRACKING API



- Endpoint Name: /shipment-status
- Method: GET
- Description: Fetch Order Shipment Status
- Response Example:

```
{ "orderId": "ORD12345", "status": "In Transit", "currentLocation": "Distribution Center", "estimatedDelivery": "2023-05-25" }
```

Rocket Seater Furniture

(API Endpoints)

Endpoint	Method	Purpose	Response Example
/products	GET	Fetches all product details	{ "id": 1, "name": "Rocket Single Seater", "price": 599, "image": "/placeholder.svg" }
/cart	POST	Adds a product to the cart	{ "success": true, "cartTotal": 1199 }
/order	POST	Places an order	{ "orderId": "ORD123", "status": "confirmed" }
/shipment-status	GET	Fetches shipment status	{ "orderId": "ORD123", "status": "In Transit", "eta": "2023-05-20" }

Rocket Seater Furniture

(Sanity Schema)

```
export default {  
  
  name: 'product',  
  
  type: 'document',  
  
  fields: [ { name: 'name',    type: 'string',    title: 'Product Name' },  
            { name: 'price',   type: 'number',   title: 'Price' },  
            { name: 'image',   type: 'image',    title: 'Product Image' },  
            { name: 'category', type: 'string',   title: 'Category' },  
  
  options: { list: [ { title: 'Sofas', value: 'sofas' },  
                    { title: 'Tables', value: 'tables' },  
                    { title: 'Chairs', value: 'chairs' },  
                    { title: 'Mirrors', value: 'mirrors' }, ] },  
  
  { name: 'description', type: 'text', title: 'Product Description' },  
  
  { name: 'stock', type: 'number', title: 'Stock Level' } ]};
```

Rocket Seater Furniture (Technical Roadmap)

Setup and Configuration

- Set up Next.js project
- Configure Sanity CMS
- Integrate necessary APIs

Day 3

Frontend Development

- Implement responsive design
- Create product listing & detail pages
- Develop shopping cart functionality

Days
4-5

Backend Integration

- Connect frontend with Sanity CMS
- Implement order processing-
- Integrate shipment tracking

Days
6-7

Testing and Refinement

- Conduct thorough testing
- Refine user experience
- Optimize performance

Day 8

Deployment and Presentation

- Deploy the application
- Prepare presentation materials

Day 9

HACKATHON_MARKET_BUILDER 2025

DAY 2 - TASK COMPLETED